Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context

Aaron Barsky, Tamara Munzner, *Member, IEEE*, Jennifer Gardy, and Robert Kincaid

Abstract— Systems biologists use interaction graphs to model the behavior of biological systems at the molecular level. In an iterative process, such biologists observe the reactions of living cells under various experimental conditions, view the results in the context of the interaction graph, and then propose changes to the graph model. These graphs serve as a form of dynamic knowledge representation of the biological system being studied and evolve as new insight is gained from the experimental data. While numerous graph layout and drawing packages are available, these tools did not fully meet the needs of our immunologist collaborators. In this paper, we describe the data information display needs of these immunologists and translate them into design decisions. These decisions led us to create Cerebral, a system that uses a biologically guided graph layout and incorporates experimental data directly into the graph display. Small multiple views of different experimental conditions and a data-driven parallel coordinates view enable correlations between experimental conditions to be analyzed at the same time that the data is viewed in the graph context. This combination of coordinated views allows the biologist to view the data from many different perspectives simultaneously. To illustrate the typical analysis tasks performed, we analyze two datasets using Cerebral. Based on feedback from our collaborators we conclude that Cerebral is a valuable tool for analyzing experimental data in the context of an interaction graph model.

Index Terms—Graph layout, systems biology visualization, small multiples, design study.

1 Introduction

Systems biology is a paradigm for biological experimentation in which researchers model biological systems by looking at the behavior of the thousands of biological entities that influence each other, rather than single biomolecules or reactions. These interactions are modeled as a graph, where the nodes represent biomolecules such as proteins and genes, and the edges represent interactions between them. This interaction graph model is used to interpret the results of experiments, and in turn experiments help biologists further refine the model.

Systems-level experimentation in cellular biology involves observing the response of cells to events by making large numbers of quantitative measurements. Examples of events are the introduction of a drug, the detection of a chemical signal from other cells, a change in environmental temperature, or the simple progression of time. A common way to observe the cell response is to measure the change in gene expression level, or abundance of proteins, in the cell across thousands of genes under a specific experimental condition. Interpreting these measurements in the context of an interaction model can help a biologist generate hypotheses about how the parts of the system influence each other.

We distinguish between two classes of interaction models available to biologists. Pathway diagrams are small directed graphs containing between ten and a few hundred nodes. These pathways show the interactions that comprise a specific biological event, such as a signaling pathway or a metabolic process. However, they provide a poor substrate for the hypothesis discovery process that drives systems biology. By limiting their representation to a small set of canonical biomolecules and interactions, the possibility of discovering new components of the process or interconnections with other processes is eliminated. Thus, systems biologists often prefer to work with the second class of data: larger undirected graphs that contain all the interactions between thousands of nodes. Graph models such as these require more effort to interrogate, but ultimately yield more novel biological insights. For example, by combining quantitative data and these large graph models, systems biologists have made discoveries ranging from the activities of new regulatory and signaling proteins, to the identification of similarly-behaving cliques that are predictive for the metastasis of cancerous tumors, to network modules that govern the aging process.

Information visualization can play an important role in the iterative model refinement process. In this domain, visualization is typically used for hypothesis generation, not hypothesis verification. Visually displaying the gathered quantitative measurements in the context of the graph model supports the hypothesis discovery process by allowing researchers to spot trends or subnetworks of interest. Testing these hypotheses then takes months or years of slow and expensive lab work.

We began a collaboration with a group of systems biologists exploring the human immune response. The primary contribution of this paper is the design of Cerebral, a new visualization tool that supports faster and richer hypothesis discovery for this group of immunologists in particular, and other systems biologists who need to see data from multiple experimental conditions in the context of an interaction graph model in general. We embarked on an iterative design process to understand the visualization needs of these immunologists that included interviews about their previous workflow and feedback from them on successive interactive prototypes.

While there are many graph layout and display systems that ably represent generic graphs [14, 16, 39], and even several aimed at biologists [36], none meet the targeted needs of our collaborators. We identified two visual requirements not currently met by existing exploration systems. First, the graph layout must use biological metadata to position nodes in biologically meaningful ways. A secondary contribution of this paper is a graph layout algorithm that incorporates biological metadata. (The existence of a tool using the layout algorithm was announced in a short Application Note [6], but algorithmic details were not given.) The second requirement is that the system must be able to simultaneously represent data gathered from multiple experiments, because the comparison between two or more experiments overlaid on a graph is a common analytical step.

In this paper, we will first discuss the workflow of our biologist collaborators, who used some existing visualization tools but undertook significant manual intervention to create the visual representations required for their tasks. From these processes, we extract information design decisions and contrast alternative methods for viewing

- *Aaron Barsky and Tamara Munzner are with Department of Computer Science, University of British Columbia, E-mail: {barskya,tmm}@cs.ubc.ca.*
- *Jennifer Gardy is with Centre for Microbial Diseases and Immunity Research, University of British Columbia, E-mail:jennifer@cmdr.ubc.ca.*
- *Robert Kincaid is with Agilent Labs, Agilent Technologies, E-mail: robert_kincaid@agilent.com.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.

the quantitative data. After reviewing the related work, we present Cerebral, a new system designed to meet our collaborators' need to interactively explore experimental data in the context of a graph model. We then present an immunology-specific scenario in which Cerebral is used to investigate the protective effects of a new therapeutic molecule, contrasting an earlier manual analysis of this data with the improved Cerebral analysis. We conclude with a more general example in which Cerebral is used to examine the passage of time in budding yeast, the first time that a visual approach has been used to explore this dataset.

2 Immunology workflow

Our immunologist collaborators use a systems biology approach to investigate human responses to bacterial infection, as well as the effects of novel therapeutic compounds on these responses. Their ultimate goal is to understand and be able to predict host responses, as well as identify therapeutic compounds that modify the immune response. Therapeutic compounds could help to resolve bacterial infections while minimizing potentially harmful side effects of the immune response, such as inflammation and septic shock.

In a typical experiment, cells are divided into a control and a treatment group. The treatment group is given a candidate therapeutic compound and then both cell populations are exposed to a simulated bacterial infection. Expression levels are measured using microarrays or other measurement technologies to determine how each gene in the cell is responding to the infection. Often a time series experiment is done, where an assay is performed at several time points to investigate how the immune response progresses.

The collected data is overlaid onto an interaction graph that models the immune response, derived from databases of known biomolecular interactions. Although tens of thousands of genes are typically measured, the immunologists do not usually work with an interaction graph model covering that entire dataset because of its overwhelming complexity. They instead consider simplified graphs of only the most interesting genes, ranging from the few dozen involved in some specific process to the few thousand involved in immunity.

Undirected edges represent the chain of interacting proteins which propagate a signal from infection-detecting proteins at the surface to the nucleus of the cell. Directed edges represent the binding of proteins to nuclear DNA, which either activates or represses the expression of genes in response to the infection signal. This ultimate response is of greatest interest to our collaborators, as it is the nature of the genes responding at this stage that determines whether the immune response will clear the infection without engaging any harmful inflammatory mechanisms.

As an example, we consider the results of an experiment previously published by our collaborators [30]. Human monocytes, a type of white blood cell associated with immunity, were stimulated with lipopolysaccharide (LPS), a molecule that can mimic the effect of bacterial infection. One batch of cells was treated with the candidate therapeutic compound LL-37, while one batch was left untreated. The gene expression level was measured using microarrays at four time points for each of these two conditions.

The figure showing this data in the published paper [30] is reprinted here as Figure 1. Although the biological graph viewer Cytoscape [33] was used to make Figure 1, creating this figure took several hours because significant manual intervention was required. Similar analyses using larger process graphs have taken days to construct.

In Figure 1, the left hand graph depicts the signaling cascade that begins with the detection of LPS by Toll-Like Receptor 4 (TLR4), proceeds through a series of intermediates, and ultimately ends in the regulation of several immune response genes. In order to create a layout reflecting the location within the cell of these biomolecules, as found in many textbooks and publications, the biologists positioned each node in the graph by hand. They thus used a very simple TLR4 graph model with only 66 nodes.

Figure 2 shows the same data displayed in our exploration tool Cerebral. Using a completely automatic algorithm, a larger TLR4 network model with 91 nodes is laid out in only a few seconds. Cerebral uses existing knowledge of where a biomolecule is found in a cell to position nodes in a biologically relevant position. The target nodes of directed protein-DNA edges are placed at the bottom of the diagram, in a layer representing immune response outcomes. The Cerebral layout groups these outcome nodes according to known biological function, enabling the biologists to easily categorize the nature of the immune response to the bacterial stimulus in the presence or absence of the therapeutic LL-37 compound.

Figure 1 also features small multiple [38] views, with each mini-graph colored according to the expression level of a gene at a specific time point. According to biological tradition, genes whose expression was significantly increased are colored red, while decreased expression levels are signified by green. As Cytoscape only loads a single experimental condition at a time, the multiple views were created one at a time by coloring the graph according to each of the eight experiments, taking a screenshot of each, and then assembling the results into a composite figure. In contrast, the small multiple views shown in Figure 2 fully support interactive exploration, with linked navigation and brushing across all windows. The coloring in the main window was chosen by clicking in two of the small multiple windows to show an automatically computed difference between those conditions.

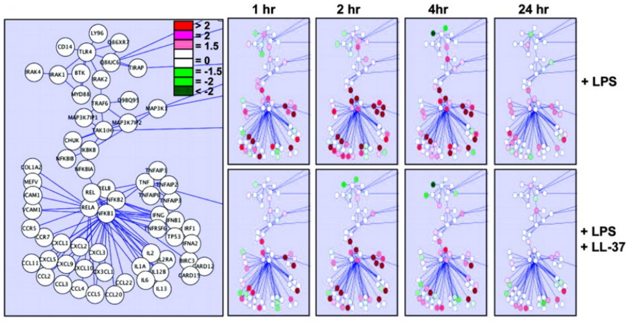


Fig. 1: Original Cytoscape analysis with manually laid out graph and manually created small multiple views, from [30].

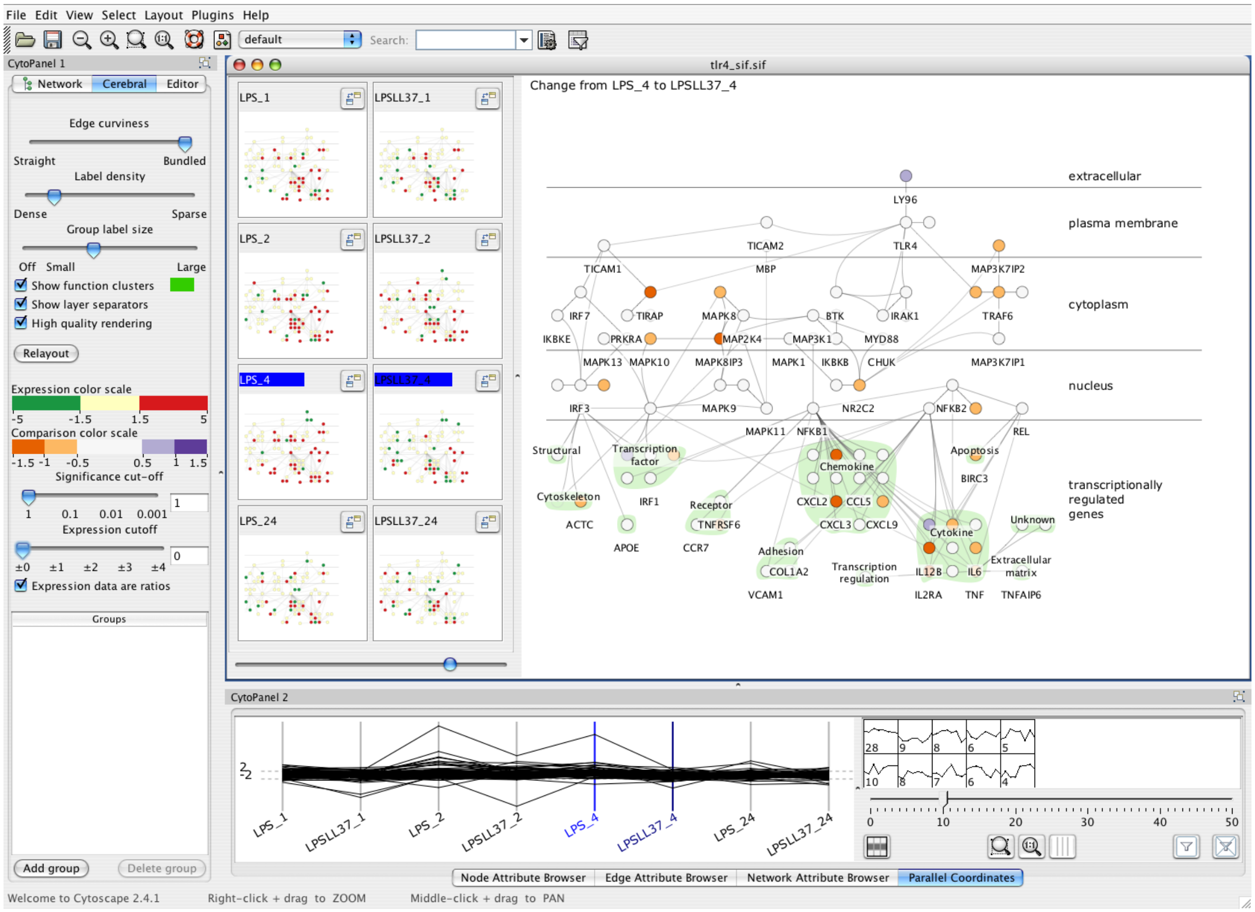
3 Cerebral design decisions

When designing an automated system to display multiple experimental conditions in the context of an interaction model, we considered several alternate data representations. We now examine our design choices.

3.1 Graph layout with biological context

Graph layout quality has historically been evaluated with a number of heuristics such as edge length, crossings between nodes and edges, and uniformity of node distribution. While many graph viewers [2, 7, 33] implement several layout algorithms that are favorably evaluated using these heuristics, none of these layout algorithms were acceptable to our immunologist collaborators when applied to their biological interaction graphs.

The nodes in biological graphs represent physical compounds in a cell that are separated by physical membranes, creating compartments defining their subcellular location. Biologists typically diagram biological processes with a stylized cross-sectional view of the cell according to this subcellular location: nodes corresponding to the outermost membrane layer of a cell are placed at the top of the graph, nodes that are found in the innermost nucleus are placed at the bottom, and the remaining nodes are placed in the center of the graph arranged neatly to show the step-wise series of interactions that occur as a signal moves from membrane to nucleus. Layout algorithms that only use graph topology to position nodes would place some nodes near each other that are always positioned in separate compartments in hand drawn immunology diagrams, causing confusion and extra interpretation effort for the biologist. Cerebral restricts the placement of nodes to these subcellular location layers, with each layer representing a distinct membrane-bound biological compartment in the cell.



☑ Show function clusters

☑ Show layer separators

☑ High quality rendering

☑ Expression data are ratios

Significance cut-off: 1

Expression cutoff: 0

Fig. 2: The Cerebral display of the TLR4 graph (V=91, E=124) with associated LPS and LPS+LL-37 time series. The small multiples show an overview of all 8 experimental conditions. The most noticeable differences between the LPS and the LPS+LL-37 condition occur at hour 4. By selecting the hour 4 conditions, the main window shows the computed difference between the two conditions.

Furthermore, the biologists' assessment of what constitutes a good layout varies depending on the nature of the biomolecules involved. In the undirected portion of the graph, which comprises protein-protein interactions that propagate a signal from membrane to nucleus, they wish to see the network structure so that they can follow the signaling cascade. Thus for this section of the graph, it is important to minimize edge crossings, even if it places interacting nodes somewhat far apart. In contrast, for the directed portion of the graph, representing the genes whose expression was altered in response to the signaling cascade, the biologists want to see the nodes grouped tightly by function, even at the expense of not being able to clearly see the interactions between them. Translating these desires into automated graph layout requires an algorithm that uses metadata associated with the nodes, in addition to the direct graph structure, for node placement. Positioning nodes according to biological meta-data defines a semantic substrate [34] so that node position reveals biological function. We wrote a simple simulated annealing-based graph layout algorithm that uses node metadata to guide node placement.

3.2 Small multiple views for multiple conditions

Cerebral uses small multiples [38] to simultaneously display multiple experimental datasets. Each small multiple contains a complete copy of the interaction graph with the same spatial layout, but with different coloring according to the experimental data it is displaying. Our design target was to handle from two to a few dozen gene expression conditions, and from 50 to 3000 nodes in the interaction graph.

One obvious alternative to multiple small views would be a single changeable or animated view, where the color coding changes over time rather than being distributed over space [33, 32]. Comparing something visible with memories of what was seen before is more difficult than comparing things simultaneously visible side by side [31]. Thus, the limitations of human memory make comparing the few dozen conditions of our design goal through animation quite difficult [40]. Although small multiples would not scale to hundreds of conditions, they handle the current usage of 8-10 easily and will certainly accommodate the projected usage of few dozen conditions.

A second alternative is to embed a glyph, such as a line graph or heat map, near or within the node itself [24, 32, 41]. While embedded glyphs provide good detail when zoomed in for a local view, they become indistinguishable when zoomed out for a global view of graphs larger than a few dozen nodes. The biologists often need to see such a view, as it more readily allows for the identification of interacting genes/proteins whose expression behaves similarly across several conditions. Thus, glyphs would not be appropriate in this domain.

Saraiya *et al.* [32] evaluated four approaches to integrating graph and time series data, comparing one versus two views and slider-controlled animation versus embedded glyphs. While they used 10 time series data points, in a good match for our problem domain, their graph contained only 50 nodes. They found many tradeoffs between task type, speed, and accuracy. Our design can be considered an attempt to combine the strengths of the four different interfaces they studied into a single interface for a problem where the tasks are complex, accuracy outweighs raw speed, and the graph is large.

3.3 Parallel coordinates and clustering for data-driven exploration

Cerebral's main views focus on the interaction graph model of the biological system or process of interest. We also provide a data-driven view and tools to help suggest areas for exploration. Each experimental condition corresponds to an axis in a parallel coordinates view. Each biomolecule, or node, in the graph is represented as a line that crosses each of these axes at points corresponding to that gene/protein's expression level under that condition, resulting in what the biologists refer to as an expression profile.

A frequent analysis of expression profile data involves clustering the profiles to identify genes/proteins that behave similarly across conditions, implying that they have related functions or are regulated by the same set of proteins. New and enhanced clustering algorithms are developed each year [42], but provide notably different results. Many of these methods are sensitive to noise, and the measurement technologies employed in systems biology often produce highly noisy results. Furthermore, the complex internals of a clustering algorithm – most often based on statistical rather than analytical decisions – make it hard for a biologist to understand and trust the clusters.

Despite these weaknesses, automatically detected clusters can be complementary to the visualization of the interaction graph in directing the biologist towards interesting groups of proteins to explore. The potential for uncovering related or novel functions and regulatory mechanisms is valuable, as is the increased confidence in the statistically-generated cluster assignments if the clustered genes are found to biologically interact with each other.

Because our immunologist collaborators do not consider any current clustering algorithm to be a clear winner, we implemented simple k-means clustering as a proof of concept. It would be straightforward to substitute a more sophisticated clustering algorithm into Cerebral. The focus in this paper is on the design choices for the multiple coordinated views [4], and the layout algorithm in the graph view.

4 RELATED WORK

Many visualization systems aimed at biological problems follow the well-established practice of using multiple linked coordinated views. For example, analysis tools such as SpotFire and HCE [23] provide a rich set of statistical tools including scatterplots, parallel coordinate views, and heat maps. However, they do not contain a biomolecular interaction graph view.

Several previous tools do allow visualization of gene expression data from a single experimental condition on a biomolecular interaction graph, including Cytoscape [33], Visant [22], GeneSpring, BiologicalNetworks [3], and GenMapp [10]. However, they are limited to visualizing data from a single experimental condition on the graph at a time and do not support automatically positioning nodes according to biological metadata. Although Cytoscape does have a graph layout algorithm that incorporates biological context, it requires manual intervention.

Interviews revealed that our immunologist collaborators were completely unwilling to experiment with layout parameters for graph drawing algorithms to obtain optimum performance. We required a robust algorithm that works well with a range of different biological datasets, without the need to change any parameters.

We thus ruled out the class of force-directed placement techniques. The straightforward approaches do not adequately support layers and groups [16]. Several attempts have been made to extend force-directed placement to accommodate grouping, for example Fruchterman and Reingold [17] use repulsive walls to contain nodes within an external boundary and Genc and Dogrusoz [18] use mobile internal walls to separate the graph into compartments. However, these methods suffer from fragility. They require parameter adjustment to work across a range of datasets. As the graphs scale up in size, it becomes a problem to balance forces. If the repulsive forces of the walls are too large, the nodes are pushed strongly away from the walls and cluster in the middle. If the wall forces are too low, all of the nodes cluster against the walls. Moreover, they have only been shown to work on small datasets of less than 100 nodes.

In contrast, Dwyer *et al.* have proposed the powerful IPSep-CoLa constraint-based approach whose running time scales well to very large graphs [14]. However, this method also requires considerable parameter tweaking to work well in this application domain. In particular, adjusting the ideal edge length parameter has considerable effect on the visual layout quality. Even after multiple rounds of personal communication with the authors, we did not obtain competitive results. Moreover, the complex algorithm is quite involved to implement, and has more power than our problem requires.

We thus chose to create a new layout algorithm that was as simple as possible to handle layering and grouping, fast enough to handle datasets of up to several thousand nodes, and would work on a range of datasets without the need for parameter changes. It is based on simulated annealing, making it straightforward to implement. Previous approaches using simulated annealing for graphs were limited to roughly 100 nodes because they required $O(V^3)$ time[11, 27, 25, 26]. We use a discretization-based framework based on a uniform grid [1] to lay out graphs with thousands of nodes in a few minutes. Tunkelang [39] also used a uniform grid to accelerate the evaluation of edge crossings when drawing undirected graphs, but assumed that edges would be short. In our approach, we use a modified version of the uniform grid that remains efficient even as edge length and expected number of crossings increases, thereby allowing a more global search of the configuration space.

We announced an earlier version of Cerebral in a short Applications Note [6] to speed its adoption by the bioinformatics community. That version featured the layout described here, but did not support the analysis of multiple conditions of gene expression data. The note did not provide any details of the layout algorithm, and did not provide a rationale for any of our visual encoding choices.

5 CEREBRAL VIEWS

The Cerebral interactive analysis system has three views of the data. The primary view shows a large representation of the interaction graph model. The user interactively chooses which single experimental condition, or computed difference between two experiments, controls the coloring of the graph nodes. A side panel shows small multiple views of the experimental conditions, with each view containing a copy of the graph model colored according to the condition data. Finally, a data-driven parallel coordinates view at the bottom shows the experimental condition data directly, visually encoded with spatial position.

5.1 Graph model views

Both the main view and the small multiple views use the biological graph as the main visual substrate.

5.1.1 Graph layout

We used simulated annealing [11] to lay out the graph using biological annotations to guide the placement of the nodes. Simulated annealing is a search strategy that repeatedly tests new configurations of a graph layout, keeping configurations that improve the quality of the graph. Worsening configurations are kept according to a temperature dependent probability function to avoid local minima in the search. The probability that a worsening change will be accepted declines by modifying the temperature with a cooling schedule. To complete our customization of simulated annealing graph layout for biology, we now define the initial state, how new configurations are chosen, how the layout quality is evaluated, and the cooling schedule.

The initial graph layout is a random distribution of the nodes. Each node's y coordinate is restricted to a layer, based on the subcellular localization annotation associated with the node. Nodes without annotation can be placed freely throughout the diagram or contained within a layer at the user's discretion. A new graph configuration is generated by selecting a single node at random and moving it to a new random position within its allowed layer.

According to our discussions with the immunologists, we evaluate the quality of the node in its new position depending on the node's function. Most nodes are evaluated based on edge length, edge crossings, and node-edge crossings. Nodes which are response proteins are evaluated primarily on distance to other response proteins sharing the same biological function, and then only weakly on edge length, edge crossings, and node-edge crossings. The classifications of nodes as response proteins and the assignment of biological functions are provided as metadata to our tool from biological databases. We weight our scores as follows: the unit weight is an edge length of 1, edge-edge crossings have weight 3, node-edge crossings have weight 9, and biological function grouping has weight 90, which encourages grouping even at the expense of node and edge overlaps. These settings were chosen by visually inspecting several graphs and choosing parameters that produced nice layouts. Though the specific choice of parameters is somewhat arbitrary, they provide good visual results for a wide range of interaction graphs containing 50 to 10,000 nodes. Further, the algorithm is robust to these parameters, continuing to give good results when they are modified by 50% to 200%. They are thus fixed in Cerebral, and do not require user tweaking.

Finally, we follow a geometric cooling schedule with a temperature decrease of 0.6 per cooling step, with 30 cooling steps, and $50V$ new configurations per cooling step, based on typical values from previous simulated annealing algorithms.

Profiling a simple implementation showed that over 98% of the algorithm running time was spent testing for edge intersections. We therefore employed the optimization technique of dividing our layout space into a uniform grid [1] and restricting possible node placements to grid centers. Each square of the grid stores a count of the number of edges that pass through the square. The squares an edge passes through are rapidly determined by using an extension of Bresenham's line drawing algorithm [8] modified to include corners. We update the grid each time a node is moved. In our evaluation function, we approximate edge crossing counts by accumulating edge counts from all grid squares that an edge passes through. We assume that all these edges cross our test edge; that is, we are computing edge crossings at the resolution of the grid and never perform an expensive floating point line intersection test. The error term in Bresenham-style algorithm keeps track of how far the edge is from the center of the square, giving us a fast way to check for node-edge intersections.

Placing nodes at grid centers has the added benefit of preventing node overlaps, leaving space for labels, aligning nodes into even lines, and assuring a compact layout by keeping nodes within the boundary of the grid.

The full details of this algorithm and its pseudocode are available [5], but are not included in this design study because of space constraints. After optimization, we were able to lay out small graphs of up to 100 edges in under 10 seconds, medium graphs of up to 500 edges in under 30 seconds, and large graphs of up to 2000 edges in under 5 minutes on a 3GHz Pentium running Linux with 2GB of RAM.

5.1.2 Graph interaction

Cerebral supports mouseover highlighting in the graph views, where the node under the cursor is highlighted in red and its graph-theoretic neighbors one hop away are highlighted in orange.

Cerebral allows users to interactively drag nodes around to override the automatic layout, and to rerun the layout after pinning down the position of an arbitrary set of nodes. This allows users to manually build a skeleton of important nodes, or incrementally layout the graph after refining the interaction model.

The graph views support panning and zooming. The label drawing algorithm guarantees that labels do not overlap at any zoom level, and the density of labels is interactively controllable via a slider. Labels are always legible: they are drawn at a fixed size in screen space, so their size varies in world space in accordance with the zoom level. We use a greedy algorithm to draw labels, using a bitmap to keep track of occupied screen space, and only drawing a label if its bounding box does not intersect any previously drawn one. We draw the label under the cursor first, then its graph-theoretic neighbors, then any nodes selected by the user, and then traverse the list of all nodes sorted by degree.

5.1.3 Using color to overlay data

We overlay gene expression or other quantitative measurements on each node of the graph by coloring the node. The appropriate color scale depends upon the reliability of the source data. Certain technologies provide accurate measurements for which a continuous gradient color scale is appropriate. Other faster and cheaper technologies will have large error bars associated with each measurement, for which a binning color scale is more appropriate. Other measurement processing pipelines will include a machine learning algorithm that outputs a simple binary value indicating whether protein level is significantly altered under the experimental condition, for which a simple two-color scheme is appropriate.

To allow maximum flexibility, our color scale editor allows the user to assign measurement value intervals to fixed or gradient color mappings. To simplify color scale creation, the editor includes a gallery of useful color scales created with ColorBrewer [19]. We provide both the red-green color scheme traditionally used for gene expression data, and a colorblind-friendly orange-purple color scale.

5.1.4 Comparing conditions

We provide a comparison view to show how gene/protein levels change between two conditions, A and B. If the data measurements are ratio values, then the difference is also shown as a ratio with condition A chosen as the new baseline. The main view colors each node as the ratio of B versus A computed as $C_x = (B_x - A_x)/|A_x|$. In the case where the data are absolute measurements, then the main view shows each node color coded by the simple computed difference $C_x = B_x - A_x$. A separate user-defined color scale is used for the comparison view.

5.2 Parallel coordinates view

Whereas the small multiple views show a large number of nodes over a small number of experimental conditions, the parallel coordinates view can focus on a small number of nodes over a large number of conditions. Each experimental condition maps to an axis in the parallel coordinates view. Each measured gene or protein maps to a line.

Outliers stand out in the parallel coordinates view, but normal values are generally occluded by the large number of lines. We have three filtering methods to extract nodes of interest. First, selection of the parallel coordinate lines is linked to the graph views. Second, we have range filters that can be set for each axis. Finally, we provide k-means clustering to find clusters of similar expression values across conditions. The current clustering dynamically creates a set of buttons allowing the user to select or deselect all members of a cluster, each with a thumbnail glyph showing the expression profile for that cluster.

Our k-clustering algorithm uses normalized expression values and Euclidean distance as the distance measure to find linearly correlated expression patterns. The k value is user selected by a slider. A standard desktop machine was able to cluster at interactive speeds, allowing the user to easily explore various values of k.

5.3 View Coordination

All graph views support linked navigation: panning and zooming in one also moves the viewpoint in all of the others. All views support linked selection and mouseover highlighting. When the user selects items, Cerebral dims the nonselected items so that they are perceived as a background layer. Selected biomolecules can be added to user-defined groups through a right-click menu. Members of the group can later be selected by clicking the group name in the Cerebral control panel. Group membership can be shown visually using attributes such as node color, node shape, or node size.

The small multiple windows and parallel coordinate axes support linked reordering through dragging. When a window is dragged to another location the axes are reordered, and vice versa.

Clicking on the titlebar of a small multiple view changes the coloring of the main view to match the selected condition, and shift-clicking a second multiple triggers the comparison coloring.

5.4 Implementation

Cerebral is an interactive system implemented in Java 1.4.2 as a plugin for Cytoscape [33]. Cytoscape is a popular biomolecular graph editor in the biology community, which loads graphs and metadata from several standard biology file formats. The Cytoscape framework allows

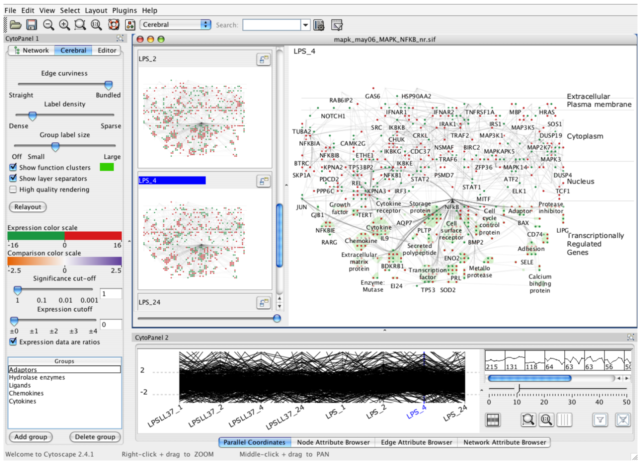


Fig. 3: The experimental LPS data shown in the context of the more complete and complex MAPK graph (V=760, E=1269).

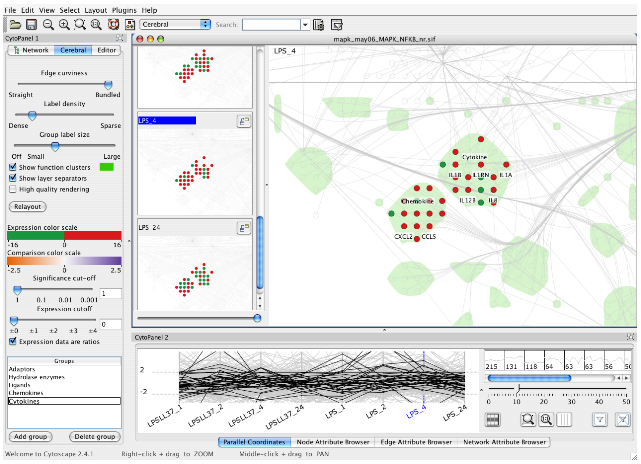


Fig. 4: Interactive selection, panning, and zooming shows the expression of cytokines and chemokines across time points in the context of the larger MAPK graph.

mapping node and edge metadata to visual appearance rules such as shape, size, and color. We replaced the standard Cytoscape renderer with our own implemented using the Prefuse toolkit [21]. Prefuse provides us with a framework for managing nodes and edges, displaying nodes, and responding to user events. We use its convex hull outlining capability to surround the functional groups in the bottom layer with colored blobs. We wrote custom code for node layout, edge bundling, edge rendering, and label positioning.

Integration with Cytoscape allows Cerebral users to take advantage of the large community of Cytoscape plugin developers. For example, the Enhanced Search plugin lets biologists select nodes with a simple query language on node metadata, and BinGO [28] creates clusters of nodes by testing for over represented gene ontology terms in the datasets.

6 Results

We present two sample sessions using Cerebral to analyze microarray experimental data in the context of a biomolecular interaction graph. We begin with an immunology-specific dataset, and then move on to a more general cellular time series dataset.

6.1 Immune response and LL-37

We now return to the LL-37/TLR4 dataset previously published by our immunologist collaborators. Figure 2 shows the data displayed with Cerebral, which performs an automatic layout exploiting the cell localization data, and displays an overview of all the datasets in the small multiple views. The outcomes of the cascade are shown at the bottom of the diagram grouped by previously known function and visually distinguished with a light green outline.

The small multiples are ordered to place each LL-37-treated time point beside the untreated condition, allowing the user to quickly spot how the peptide affects the cell's response to bacterial infection. Scanning these pairs, we see that the differences in the two conditions are most pronounced at hour 4. We note there are many more nodes colored solid green in the LPSLL-37_4 condition than the untreated LPS_4 condition. Rather than scanning back and forth between the two conditions to search for differences, we have Cerebral compute the direct difference between the two hour 4 conditions. The difference view is colored with an orange/purple color scale.

Most of the significantly changed nodes in the difference view are colored orange, indicating that the expression levels are reduced in the presence of LL-37. Furthermore, it is easy to see that the response proteins that show such changes primarily belong to the cytokine and chemokine functional categories. Many of these are implicated the harmful inflammatory side effects of the immune response to bacterial infection, thus it appears that the protective LL-37 molecule is working to minimize these side effects.

Recreating the same data views as the original paper took seconds in Cerebral, compared with many hours of manual manipulations. Moreover, the subcellular localization in Cerebral exposed a longstanding error in this well-studied small graph that was not apparent in previous automatically produced layouts using generic force-directed layout algorithms. A long edge from the top `extracellular` layer crossed all three middle compartments to end in the bottom `transcriptionally regulated genes` layer. After seeing the layout, the immunologists were prompted to double-check the annotation of the nodes connected to the long edge, realized that it was incorrect, and refined the interaction graph model in their database.

The automatic layout and interactive exploration capabilities of Cerebral allows the immunologists to productively use more complex and complete graph models. Figure 3 shows the same LPS gene expression experiment data overlaid on the MAPK model graph, a superset of the TLR4 graph containing 1269 edges and 760 nodes. While the overview of the complete graph is quite complex, the interactive selection and navigation allow the immunologists to easily explore across multiple time points, as shown in Figure 4.

6.2 Yeast cell cycle: Time series analysis

This example uses publicly available gene expression data from a time series study of the cell cycle of yeast published by Spellman et al. [35]. This expression data was combined with a yeast cell cycle protein interaction graph constructed by de Lichtenberg et al. [12] and available from the Cell Circuits database [29]. Cellular location information was obtained from the Yeast Protein Localization Server [13]. Data from these sources were then combined to form an integrated data set suitable for analysis by Cerebral. Though the Spellman study investigated how gene expression varied according to cell cycle phases, they did not examine the data in the context of a biological network. This example illustrates that the network context shown with Cerebral allows clear and fast detection of correlated expression changes.

Figure 5 shows the initial Cerebral display generated when we first load the data. We simultaneously view the entire set of expression data across all time points using the parallel coordinate display as well as the small multiples display. The automated Cerebral layout of protein interactions is arranged by cellular location as well as by interactions. We have clicked in the `cdc050` small multiple window, so the main view shows the coloring for that condition.

As we are not starting with a particular hypothesis, we begin exploration by using the data-driven parallel coordinates view. We interactively adjust the k-means slider until we see an interesting correlation pattern in the cluster glyphs. We click a sinusoidal pattern in the lower-right thumbnail cluster buttons, since this pattern is suggestive of cell cycle phases. In Figure 6 the parallel coordinate display clearly shows the sinusoidal nature of the selected cluster across the time series, indicative of its involvement in the phases of the cell cycle.

In fact, we see that six of the molecules in this cluster correspond to histone genes [37]. The graph view quickly and clearly conveys the fact that these molecules interact with one another and are active in the nucleus. Figure 7 shows the result of selecting this smaller subset for closer inspection by dragging out a box in the main view. We now see even more striking correlated behavior in the parallel coordinates plot. Viewing the small multiples, we can scan the time series in the graph context and see how the cluster shifts as a unit from showing green under-expression to red over-expression. A trained biologist will immediately recognize this pattern as following the cell cycle phases. We note that while the parallel coordinate view can represent this correlated behavior, it cannot show the relationship between the proteins. The small multiple graph views show such correlation qualitatively, but they do not present as precise an analytical display as the parallel coordinate plot. By coordinating both views simultaneously, the user can visualize and compare views for more complete analysis.

This analysis session shows how the simple interaction graph could be extended to include temporal effects. We see how histone levels rise and fall in time with the progression of the yeast cell cycle.

6.3 User Response

We gathered feedback from our immunologist collaborators on the design of Cerebral as they used a succession of interactive prototypes. We began by attacking the problem of biologically-guided layout, with a prototype that had only a single interactive graph view. After a period of refinement, we made this early single-view version of Cerebral publicly available in February 2007. The response from the bioinformatics community was very encouraging: we have heard from many groups who have used it, and three published biology papers include figures created with Cerebral. One is from our collaborators [9], and two are by other researchers [15, 20]. We note that one of the latter [20] explicitly mentions Cerebral in the methods section of the paper, showing that it was considered integral to their analysis methodology as opposed to simply being used for presentation.

We then continued on to the problem of handling multiple experimental conditions, again refining the design through feedback on prototypes. The final version of Cerebral tool documented in this paper has been enthusiastically embraced by our immunology collaborators. They have integrated Cerebral as the visualization front end for InnateDB (http://innatedb.ca), their hand-curated immunology interaction database. Full source code and binaries for this version were made publicly available as Cerebral v.2.0 in May 2008 at http://www.pathogenomics.ca/cerebral.

We expect further user feedback now that the tool has been publicly released to a larger audience. Cerebral was evaluated many times by our collaborators during the iterative design process as we worked with them to identify and satisfy their usability and visualization needs. However, this close working relationship means that their favorable final evaluation is not impartial, so formal user testing with more neutral groups of biologists would be interesting future work.

7 Future Work and Conclusions

Our tool met the design goal of analyzing a few dozen conditions with a few thousand nodes, but has the limitation that it does not scale to significantly larger numbers of conditions and graph nodes. As high-throughput biomolecule measurement technologies become more scalable and cost effective, biologists will include increasing numbers of experimental conditions in their study design. In order to support these larger study designs, we would need to improve both the visual scalability and the computational performance of the system.

Cerebral also has the limitation that the user interface only allows nodes to be arranged into an ordered layered hierarchy. Some cell compartments do not follow a linear hierarchy; for instance, mitochondria and Golgi bodies are both found in the cytoplasm. Building an input system that supports subregions within layers for organelles, and circular regions for bacterial cells, should be a straightforward extension of our underlying layout algorithm. Another interesting area of

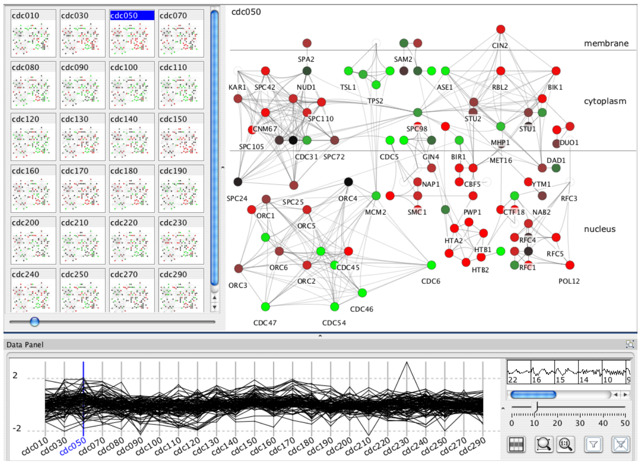


Fig. 5: Initial Cerebral layout of the yeast cell cycle data (V=104, E=417) shows expression data in a subcellular localized graph context.

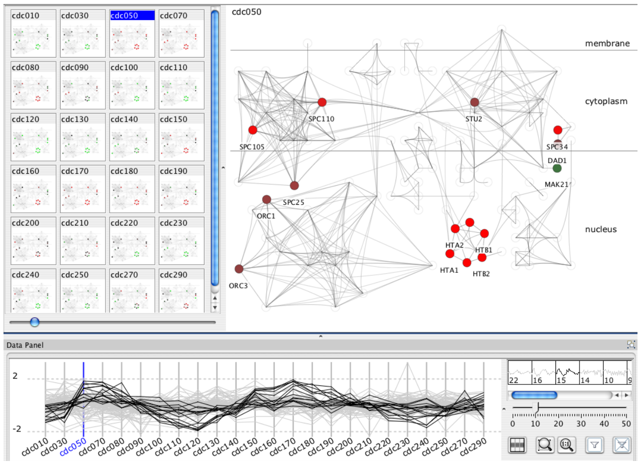


Fig. 6: We set the k-means slider to 10 clusters, and select a cluster of 15 genes that show cyclic behavior.

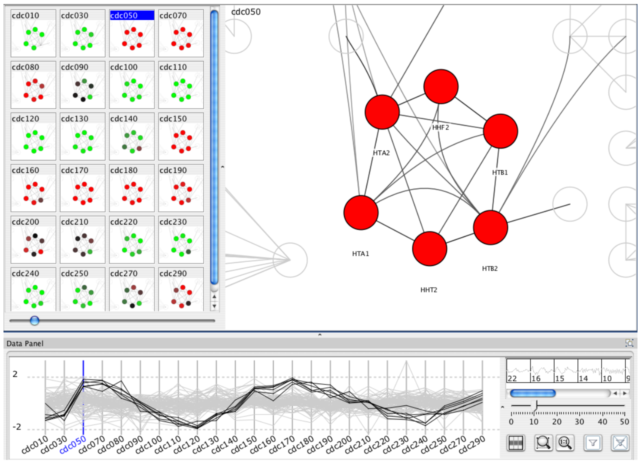


Fig. 7: Selecting the interrelated histone group (HTA1, HTA2, HTB1, HTB2, HHF2, HHT2) by dragging in the main view shows a clearer pattern of early under-expression in green, then later over-expression in red, in the small multiple views.

future work would be to identify other application domains that have graphs with linearly ordered classes of nodes and test whether this simulated annealing layout approach is an adequate solution in more generic contexts.

We have shown that overlaying experimental data on an interaction graph with Cerebral provides systems biologists an opportunity to evaluate the current biological system model, generate hypotheses, and improve and refine the model. Cerebral has data displays customized for systems biology tasks, providing an environment where the data views are familiar and the graph nodes appear in biologically sensible locations. By creating biologically meaningful graph layouts automatically, systems biologists are now able to work with much larger and more complete graphs. Interactive simultaneous viewing of multiple experimental conditions allows for more in-depth analysis of gathered data.

Acknowledgements

Barsky was supported by a grant from Agilent Technologies. We thank the members of the Hancock Lab for the time spent giving feedback on prototypes and for the LL-37/TLR4 dataset.

References

[1] V. Akman, W. R. Franklin, M. Kankanhalli, and C. Narayanaswami. Geometric computing and the uniform grid data technique. *Computer Aided Design*, 21(7):410–420, 1989.

[2] D. Auber. Tulip : A huge graph visualization framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Software*, Mathematics and Visualization, pages 105–126. Springer-Verlag, 2003.

[3] M. Baitaluk, M. Sedova, A. Ray, and A. Gupta. BiologicalNetworks: visualization and analysis tool for systems biology. *Nucl. Acids Res.*, 34(suppl_2):W466–471, 2006.

[4] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. Advanced Visual Interface (AVI '00)*, pages 110–119, 2000.

[5] A. Barsky. Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Master's thesis, University of British Columbia, Dept. Computer Science, Vancouver, June 2008.

[6] A. Barsky, J. L. Gardy, R. E. Hancock, and T. Munzner. Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. *Bioinformatics*, 23(8):1040–1042, 2007.

[7] B.-J. Breitkreutz, C. Stark, and M. Tyers. Osprey: a network visualization system. *Genome Biology*, 4(3):R22, 2003.

[8] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[9] K. L. Brown, C. Cosseau, J. L. Gardy, and R. E. Hancock. Complexities of targeting innate immunity to treat infection. *Trends in Immunology*, 28(6):260–266, 2007.

[10] K. Dahlquist et al. GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet.*, 31:19–20, 2002.

[11] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Trans. on Graphics*, 15(4):301–331, 1996.

[12] U. de Lichtenberg, L. Jensen, S. Brunak, and P. Bork. Dynamic complex formation during the yeast cell cycle. *Science*, 307(5710):724–727, 2005.

[13] A. Drawid and M. Gerstein. A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome. *J. Mol. Biol.*, 301:1059–1075, 2000. `http://bioinfo.mbb.yale.edu/genome/localize/`.

[14] T. Dwyer and K. Marriott. IPSep-CoLa: An incremental procedure for separation constraint layout of graphs. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis '06)*, 12(5):821–828, 2006.

[15] M. D. Dyer, T. M. Murali, and B. W. Sobral. The landscape of human proteins interacting with viruses and other pathogens. *PLoS Pathog*, 4(2):e32, Feb 2008.

[16] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. Graph Drawing 1993 (GD03)*, LNCS 894, pages 388–403. Springer, 10–12 1994.

[17] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[18] B. Genc and U. Dogrusoz. A constrained, force-directed layout algorithm for biological pathways. In *Proc. Graph Drawing 2003 (GD'03)*, LNCS 2912, pages 314–319. Springer, 2004.

[19] M. A. Harrower and C. A. Brewer. ColorBrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[20] L. He et al. The glomerular transcriptome and a predicted protein-protein interaction network. *J. Am. Soc. Nephrol.*, 19(2):260–268, 2008.

[21] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. ACM CHI '05*, pages 421–430, 2005.

[22] Z. Hu, J. Mellor, J. Wu, and C. DeLisi. VisANT: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, 5(1):17, 2004.

[23] B. S. Jinwook Seo. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35(7):80–86, 2002.

[24] B. Junker, C. Klukas, and F. Schreiber. Vanted: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, 7(1):109, 2006.

[25] M. Kato, M. Nagasaki, A. Doi, and S. Miyano. Automatic drawing of biological networks using cross cost and subcomponent data. *Genome Inform.*, 16(2):22–31, 2005.

[26] K. Kojima, M. Nagasaki, E. Jeong, M. Kato, and S. Miyano. An efficient grid layout algorithm for biological networks utilizing various biological attributes. *BMC Bioinformatics*, 8:76+, March 2007.

[27] W. Li and H. Kurata. A grid layout algorithm for automatic drawing of biochemical networks. *Bioinformatics*, 21(9):2036–2042, May 2005.

[28] S. Maere, K. Heymans, and M. Kuiper. BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21:3448–3449, 2005.

[29] H. C. Mak, M. Daly, B. Gruebel, and T. Ideker. CellCircuits: a database of protein network models. *Nucl. Acids Res.*, 35:D538D545, 2007. `http://www.cellcircuits.org/`.

[30] N. Mookherjee et al. Modulation of the Toll-like receptor-mediated inflammatory response by the endogenous human host defence peptide LL-37. *J. Immunol.*, 176:2455–2464, 2006.

[31] M. Plumlee and C. Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *Proc. ACM Trans. on Computer-Human Interaction (ToCHI)*, 13(2):179–209, 2006.

[32] P. Saraiya, P. Lee, and C. North. Visualization of graphs with associated timeseries data. In *IEEE Symp. Information Visualization (InfoVis 2005)*, pages 225–232, 2005.

[33] P. Shannon et al. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13:2498–2504, 2003.

[34] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Trans. Visualization and Computer Graphics, (Proc. InfoVis 2006)*, 12(5):733–740, 2006.

[35] P. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[36] M. Suderman and M. Hallett. Tools for visually exploring biological networks. *Bioinformatics*, 23(20):2651–2659, 2007.

[37] A. Sutton, J. Bucaria, M. A. Osley, and R. Sternglanz. Yeast ASF1 protein is required for cell cycle regulation of histone gene transcription. *Genetics*, 158:587596, 2001.

[38] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[39] D. Tunkelang. A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, Carnegie Mellon University Department of Computer Science, June 1994.

[40] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[41] M. A. Westenberg et al. Interactive visualization of gene regulatory networks with associated gene expression time series data. In *Visualization in Medicine and Life Sciences, Visualization and Mathematics*, pages 293–312, 2008.

[42] K. Yeung, M. Medvedovic, and R. Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, 2003.